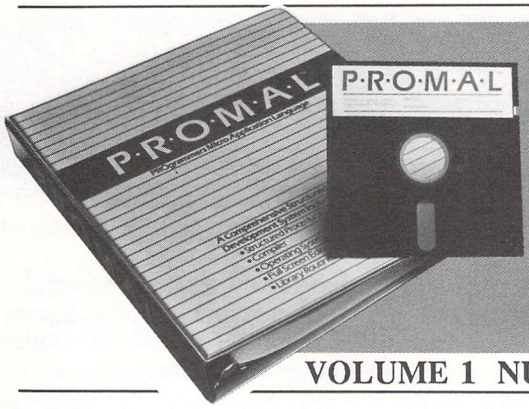


PROMAL™ NEWS

PROgrammer's Micro Application Language

VOLUME 1 NUMBER 1 April-June, 1986

WHAT'S NEW(S)

PROMAL NEWS Dedicated to PROMAL User Community Debuts with First Quarterly Issue

Welcome to PROMAL NEWS, the newsletter for PROMAL programmers. Our intention is to publish PROMAL NEWS quarterly to provide you with news, programming support, new product and upgrade notices, User Group news, and information on the latest additions to the PROMAL Public Domain Library.

This Newsletter is **your forum** (as well as ours) and we welcome your comments and suggestions about the newsletter's format and content. **Tell us what you would like to see and feel free to contribute material.**

We at SMA continue to be gratified by the enthusiastic response so many of you have given us about PROMAL, and we hope this newsletter will help you all become better PROMAL programmers.

Now for some more news.

The IBM version of PROMAL has been delayed until the end of May and will ship as a preliminary (1.9) version which will be followed by 2.0 later in the year. You will all receive a mailing about the IBM version.

The **PPDL** is growing and has some **great** programs, including a 6502 assembler. See the PPDL section on page 5 and the order form for details.

Our publisher (who is also President of SMA, Inc.) has insisted on having his own column called "The Last Word", please humor him by reading it. He has a **special offer** for those of you who are sometimes motivated by money.

And the best news of all...

PROMAL Graphics Toolbox Now Available

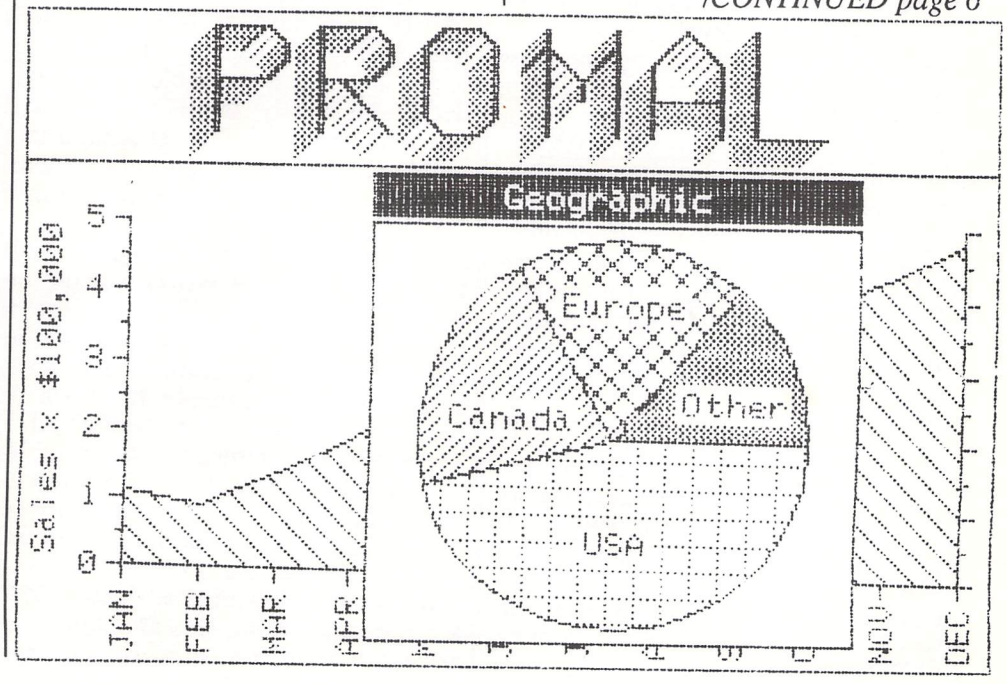
Now you can program **dazzling high-res graphics** in PROMAL. The PROMAL GRAPHICS TOOLBOX provides graphics commands for the Apple II or Commodore 64 with performance to match the rest of the PROMAL system.

The figure below shows a screen created with the GRAPHICS TOOLBOX on an Apple IIc. By adding this software package, **you can easily write programs to draw bar graphs, pie charts, function plots, and other graphic images.**

The GRAPHICS TOOLBOX for the Commodore 64 draws in 16 colors using the 320 X 200 high-resolution mode. On the Apple, graphics are done in 280 x 192 high resolution monochrome, since the Apple does not have a high-res color mode. Despite substantial differences in the underlying hardware, graphics applications written for the Commodore are highly "portable" to the Apple and vice versa.

The GRAPHICS TOOLBOX provides two add-on libraries: the Screen Graphics Drivers (SGD) and the Window Graphics System (WGS). The SGD provides high-performance "primitive" functions for drawing points, lines, dashed lines, rectangles, circles and arcs, etc., plus routines for

/CONTINUED page 6



The Last Word

(A personal letter from your friendly Publisher)

Dear Friends:

I asked the Editors to let me have a small space in the newsletter to talk one on one to you. You make it possible for us to be in the programming language software business. We want to serve you with the best possible product at a fair price.

From the feedback I get from you we seem to be on the right track with PROMAL. It is truly a value-packed product. But I've learned in the microcomputer business that it's not enough to have a great product, you've got to have great marketing to become established as a leader.

So we've got a dilemma. Since we're a relatively small company and don't have megabucks for advertising, how will we achieve the market success that PROMAL deserves? Every $10,000 spent on sales promotion is $10,000 less available for future PROMAL product development.

My solution is simple, but requires your help. We need our existing PROMAL users to convince their friends and colleagues to try PROMAL -- classic "word of mouth" promotion. If every current user of PROMAL could convince three or four other programmers to buy PROMAL we could triple our user base in just a few months, with essentially no marketing costs. This means we could invest more into accelerated PROMAL development and support.

Everyone benefits. A rapidly growing PROMAL user base means more Public Domain software, more user groups, more new PROMAL products, and more support from SMA. But, I would like to do something for each of you immediately in exchange for your efforts to promote PROMAL.

This is my offer. For each copy of PROMAL we sell as a result of your effort, we'll pay you a "commission" in the form of credits to be used against the purchase of future PROMAL products, upgrades, or Public Domain disks.

Here's how it works. You convince a friend to buy PROMAL and have him or her write "Referred by:" and your name and address on the registration form when they send it in. Immediately on receiving the registration form we will send you credit coupons based on the following schedule:

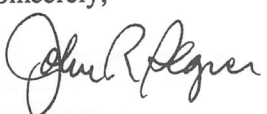
PM-320 (Developer's Version)	=$10.00
PM-220 (End User Version)	= $5.00
PM-410 (Graphics Toolbox)	= $3.00

The credit coupons are cumulative, transferable, and have no expiration date. So if you got four friends to buy PM-320s you'd get $40.00 in coupons to use anytime in the future. If you want to let another friend have a good deal, give him one of your coupons to send in with his order (a truly selfless act) or sell him one or more of your coupons at less than "face" value. You get cash and he gets a discount. Everybody wins.

We'll report in the next newsletter our top coupon winners. Who knows, you might even get offers from other users to buy your coupons.

Thank you for your support of PROMAL and I'll look forward to sending out those coupons.

Sincerely,



John R. Segner
President, SMA, Inc.

□

FIXES and UPDATES

The purpose of this column is to tell you about the known bugs in PROMAL and give you fixes or techniques to overcome them. Many software companies seem to strive to hide bug information from users, or even deny the existence of bugs even in the face of overwhelming evidence.

We think that all but the most naive users realize that all published software contains bugs, and we also think you deserve to know about bugs up front so you can deal with them. So here are the problems:

1. If you have a program which uses EXPORTs and has a FOR loop with an index variable which is declared globally, the FOR loop will not work properly. To get around this problem, simply declare your index variable as a local variable (i.e., inside the PROC or FUNC).

2. If you compile a program with EXPORTs from the Workspace on the Apple, you may get a "Can't open EXPORT file" error message at the end of the compilation. To work around this, just compile from a disk file instead.

3. On the Commodore 64 or 128 with a 1571 disk drive, the system may occasionally "hang up" while accessing the disk. Contrary to claims by Commodore, the 1571 drive is not 100 percent compatible with the 1541. We are researching this problem. In the mean time, we suggest you turn DYNODISK OFF if you have a 1571 or if you experience this symptom.

4. For formatted output of REAL numbers, if you try to output exactly the smallest value possible in a given format, it may print zero instead. For example, using a field descriptor of "#12.3R" with a value of 0.001 may display .000 instead of .001. However, all other values, such as 0.001000001, 1.001, etc. work fine. This problem can be fixed by making the following patch to the PROMAL system:

Version	Address	Was	Should be
Apple (both)	$1785	$F0 13	$EA EA
C-64 Regular	4C7F	F0 13	EA EA
C-64 GENMASTERed	4BDF	F0 13	EA EA

You can put this patch in your BOOTSCRIPT.J file. For example, for the Apple, add

```
SET 1785 EA EA
```

to the BOOTSCRIPT.J file to make the patch. For stand-alone programs generated with the developer's disk, you can use assignment statements to make the patch. For example, for a Commodore stand-alone application:

```
IF M[$4BDF]=$F0 AND M[$4BE0]=$13
   M[$4BDF]=$EA ; Patch 2.0 bug
   M[$4BE0]=$EA ; in REAL output
```

/CONTINUED page 6

PROGRAMMING TIPS & TECHNIQUES

by Bruce Carbrey, SMA, Inc.

Safe Handling With PROMAL Arrays

Programming in PROMAL after using BASIC is like getting a sports car after years of riding the bus. It's fun, fast, and will take you anywhere you want, but it's kind of scary too at first; if you don't understand what the controls do or get careless, you're likely to crash! Here are some driving tips to keep you out of the ditch.

When using arrays and strings, it is important to understand the difference between the address of an array and the value of an element of the array. Suppose you had an array declared as:

```
BYTE X [81]
```

Now suppose you wanted to read a line of text into this array from the keyboard. No sweat, you say, just use:

```
GETL X[0]
```

This will put it in the array X starting at the first byte, right? CRASH! Hope your insurance is paid up, because that won't do it! GETL, like most string-handling routines, expects the address of the first byte of the string. But typing X[0] with the brackets means "get the value of the first byte of the array", which will be whatever number (between 0 and 255) happens to be in X[0] at the time. GETL will obediently treat this number between 0 and 255 as an address and install the string somewhere in zero-page of your computer, tearing up the operating system innards in the process and probably hanging up your computer.

So how do you specify that you want the address, not the value? There are two ways. The simplest is simply to write the array name without the brackets. That causes the compiler to generate the address of the first element of the array. Therefore a correct way to input a line into the X array is:

```
GETL X
```

Okay, you say, what happens if you don't want to install the string at the very start of the array. Suppose instead you have a 2-dimensional array like:

```
BYTE PAGE [41,25]
```

with room for up to 25 lines of text in it. Say you want to get the string into the 3rd line. This will do it:

```
GETL #PAGE[0,2], 40
```

The # operator causes the compiler to generate the address of the specified element. Remember that the subscript 2 is used instead of 3 because the first element is always 0, not 1. More importantly, remember that the last element is 24, not 25! If you try to read a line into #PAGE[0,25], PROMAL will oblige you by writing into whatever happens to be in memory after the end of the PAGE array (hope you're wearing your seat belt). Also note that the optional second argument (40) after GETL is good insurance. Without it, you could enter up to 80 characters, even though the array was only dimensioned large enough for 40.

To summarize, if a PROMAL library routine (or any PROMAL subroutine for that matter) expects a string, you need to specify an address. If in doubt about how to make an address, place the # operator in front of the variable. The following are all exactly equivalent:

```
PUT X
PUT #X
PUT #X[0]
```

All three expressions will print the string which starts at the first location of the X array in memory (and is terminated by a zero byte). The following statement is not equivalent:

```
PUT X[0]
```

This statement will print only the first character of the string, because the expression will evaluate to the value of the first byte of the array X (which is assumed to be an ASCII character code). PUT is one of the few routines that can accept a single character or a string. If the argument is less that 256, PUT assumes the argument is a single ASCII character and prints it. If the argument is greater than 256, it assumes the argument is the address of a string to be printed.

With a little practice and maybe some studying of the SORT-STRING.S program on the demo diskette, you'll soon be using arrays like a master. Keep it on the road, OK? □

THE ECHO-LESS GETC

Several users have inquired as to how you can get a single character from the keyboard without having it "echoed" to the screen, as is done by function GETC. Here's a simple way:

```
FUNC BYTE GETC_NOECHO
  ; Waits for key without
  ; echoing to screen.
  ; Returns the key code.
BYTE TEMP
BEGIN
WHILE TESTKEY(#TEMP)=0
  NOTHING
RETURN TEMP
END
```

Unlike GETC, this subroutine does not display a blinking cursor while waiting. □

PRINTER <ESC> SEQUENCES

When sending escape sequences to a printer on the Commodore 64, you may need to defeat the ASCII-to-CBM upper/lower case flip normally done by the PROMAL printer driver. For example, sending 27, 76 (ESC, L) will really be sent as 27, 108 (ESC, 1). To correct this, use:

```
EXT BYTE C64PUL AT $0DF4
; $80=FLIP CASE,0=don't
...
C64PUL=0
; Turn off prtr case flipping
PUTF 27,76
; send raw ASCII codes as needed
```

□

Disk is Ridiculous!

Requiem for a Programmer

"No program is perfect."
They said with a shrug.
"The client is happy --
What's one little bug?"

But he was determined.
The others went home.
He dug out the flow chart
Deserted, alone.

Night passed into morning.
The room was cluttered
With memory dumps, microfiche;
"I'm close," he muttered.

Chain smoking, cold coffee,
Logic, deduction.
"I've got it," he cried.
Just change one instruction."

Then change two, then three more
As year followed year.
And strangers would comment,
"Is that guy still here?"

He died at the console
Of hunger and thirst
Next day he was buried
Face down, nine edge first.

And his wife through her tears,
Accepted his fate.
Said, "He's not really gone,
He's just working late."

-Original author unknown.

Submitted by David Simpson, Laurel, MD.

ED. NOTE: If any of you under-40 programmers don't understand the last line of verse six, ask an "old-timer" what it means.

LETTERS TO THE EDITOR

In future issues, we will print letters from users in this section. For this first issue, we will paraphrase excerpts from some letters we have received.

"... According to Appendix G of the PROMAL manual, up to about 32K bytes are available for loading programs on the Commodore 64, but I get an error message if I try to run a program which is larger than about 19K bytes. What gives?"

The EXECUTIVE is set up to preserve the EDITor, which takes up about 12K or so. To use the full available space, you need a boot loader which will overwrite the EDITor. Here is a general purpose boot loader which will load a specified big program, provided the program was written with OWN on the PROGRAM line:

```
PROGRAM BOOTBIG OWN
; Commodore 64 only!
; Boot program specified as the argument;
; Clobbers the EDITor and workspace.
; Does not unload programs (except EDITor).
INCLUDE LIBRARY
INCLUDE PROSYS
BEGIN
IF NCARG <> 1
  PUT CR,"BOOTBIG ABORTED: NO NAME GIVEN."
  ABORT "#cUSAGE: BOOTBIG Progname"
HIFREE=MEMLIM ; The max memory
WORG=MEMLIM   ; No workspace please
WPTR=MEMLIM
WEOF=MEMLIM
WLIM=MEMLIM
EDRES=FALSE   ; Goodbye EDITor
LOAD CARG[1]  ; Load & run big prog
IF LDERR <> 0
  ABORT "#cBOOTBIG LOAD ERROR $#h",LDERR
END
```

"...I was really disappointed to see you cheated with the number of primes to calculate in the Eratosthenes' sieve program from BYTE. Your product is really good, it doesn't need that kind of unfair tricks..."

-C.N., Mougins, France

We didn't cheat. We bent over backwards to make the competition look good. The reason the array size used in the benchmark tests (in SMA magazine ads) was reduced from 8192 to 1800 is because COMAL (not to be confused with PROMAL) could not run with an 8192 element array. So rather than disqualify COMAL, we changed the array size for all the languages to the biggest size that COMAL could accommodate (about 1800). The relative performance of the languages is still the same - with PROMAL way out in front.

"...The cursor blinks too fast. How can I slow it down on my Apple?"

For Apple PROMAL version 2.0 only:

```
SET BE83 0
```

"...The designers of PROMAL deserve as much respect as Wirth, or Kernighan and Ritchie. Your language is very good...I am also impressed with the other new features and extensions present in PROMAL 2.0...I will pass on information to LOGIC (Loyal Ontario Group Interested in Computers), the largest Apple users' group in Canada...Congratulation on a wonderful language and best wishes for continuing and growing success."

- J.M., Toronto Ontario

To you and all the hundreds of other users who took the time to write similar sentiments, we thank you. It is really rewarding to see so much enthusiasm for PROMAL. When we first introduced PROMAL, a lot of experts told us we were crazy to sell it so cheaply and more crazy to provide ongoing support. We think the experts were wrong. If you think so too, we encourage you to tell your friends and write to magazines about PROMAL. This is the most effective way you can send software publishers a message that you are more interested in quality and support than in slick ads and hype. Again, thank you for your encouragement from all of us at SMA. □

Public Domain Disks Now Available

These are public domain contributions by users. You can freely use or copy them, SUBJECT ONLY TO ANY RESTRICTIONS IMPOSED BY THE PROGRAM AUTHOR. (For example, some authors may have copyrighted their programs and disallow commercial use.)

SMA serves only as a clearing house for the PPDL (PROMAL Public Domain Library) and in no way stands behind program correctness or provides support for them. Don't even think about calling or writing us about PPDL problems.

All programs are written in PROMAL and supplied in both source and compiled form. All documentation is supplied by the authors. A documentation file accompanies most programs. A few have no documentation, but they are easily understood by a look at the source code. Hardcopy documentation is provided when the amount is large or when hand-drawn diagrams are present.

Disk #1 for the Commodore

Contents:

1. Macro assembler, by C. Martens
2. Disassembler, by Steve Vermeulen
3. Disk Fixer program, by A. Ryan. included also is a twenty-page manual describing Commodore disk structures and how to change them using the disk fixer.

The delivered product consists of a disk and summary of contents, plus the Disk Fixer documentation in hardcopy form.

Disk #2 for the Commodore

Contents:

1. Document formatter program, by David Long. It offers a powerful and well established method of word processing that begins with a manually created file containing text intermixed with formatting commands. The commands define such things as margins and page size, paragraphs and "absolute" lines of text, underlining, page headers and footers, and page numbering. The manually created file, which makes no pretense of looking like the final document, is fed into the document formatter program, which acts upon the commands and produces a printer-ready result. A file is included that, when fed through the document formatter, produces the fifteen page user's manual that is also supplied in hardcopy form.
2. Screen creator, by Rev. Mike Cargill
3. Printer control issuer, by Julia Christianson
4. PROMAL source file lister, by Garth Ingram
5. Screen creator, by W. A. Marsh
6. Graphics routines and demo, by Roger Norrod
7. Lister that includes time and date stamp, by Michael T. Veach
8. C64-to-Tandy PC2 data exchange program, by Steve Vermeulen
9. "Dumb terminal" emulation routine and demo, by Steve Vermeulen
10. KOALA touchpad support, by Erik Vigmostad
11. Counter of word occurrences in a file, by Erik Vigmostad
12. File lister for RS-232 printer, by Erik Vigmostad

The delivered product consists of a disk and summary of contents, plus the Document Formatter documentation in hardcopy form.

Disk #1 for the Apple

Contents:

1. Macro assembler, by C. Martens
2. Disassembler, by Steve Vermeulen
3. Printer control issuer, by Julia Christianson
4. PROMAL source file lister, by Garth Ingram
5. Counter of word occurrences in a file, by Erik Vigmostad

The delivered product consists of a disk and summary of contents.

Disk #2 for the Apple

Contents:

1. Document formatter program, by David Long. It is described above under "Disk #2 for the Commodore", item 1.

The delivered product consists of a disk and summary of contents, plus the documentation in hardcopy form. □

CUT HERE

FAME AND FUN FOR YOU!

The PROMAL Newsletter and Public Domain Disks give you a forum to show off your software masterpiece, see your name in print, and benefit other programmers. So come on. Send in that nifty little program or routine you've been working on, so others can marvel at your ingenuity and benefit from your experience. These contributions don't have to be big or complex; the best ideas are usually the simplest.

To submit a program for the public domain disk, send the source and object code on a disk together with a text file telling what the program does and how to use it (this may be obvious to you, but without some documentation, it won't be much help to others). Examples are great. Put a label on the disk telling what kind of disk it is (Commodore or Apple), and add your name and phone number. Enclose a note (handwritten is fine) telling us it is a submission for the PROMAL Public Domain Library and that you are the author. Sign and date it. That's all there is to it!

For articles for the newsletter, you can submit a disk with a text file. We can accept Apple, Commodore, or IBM (MS-DOS) disks. For IBM disks, please use ordinary ASCII text files (or WordPerfect, WordStar or MultiMate files), and tell us which kind of file it is.

If your article or program is selected for publication, you will get the next volume of the PROMAL PUBLIC DOMAIN DISK free! But more importantly, you will be helping the PROMAL community and getting pleasure out of seeing your labors appreciated by others. Besides, the sooner we get enough good submissions for a new disk, the sooner you can get another volume of the PUBLIC DOMAIN DISK.

Here are a few topics we'd particularly like to hear about:

1. Interrupt-driven T device support for the Apple IIc.
2. Alternate character fonts for the PROMAL GRAPHICS TOOLBOX.
3. Games
4. Utility programs

/CONTINUED page 6

/NEWS from page 1

displaying text in graphics mode (in various sizes, horizontally or vertically), saving and restoring images or parts of images, filling an area, etc. The SGD uses an (X,Y) coordinate system based on the pixels (dots) on your screen. For example, to draw a rectangle 60 pixels by 40 pixels with its lower left corner 90 pixels from the left of the screen and 50 pixels up, you would write:

```
S_MOVE 90,50
S_RECT 60,40
```

If you wanted to add a title inside the rectangle, you could just write:

```
S_TEXT "A Rectangle"
```

For maximum performance, the SGD is provided as a relocatable machine language program, occupying about 4.5K of memory (reducible to 2K if no text or pattern-filling is required). The SGD can be used alone or in conjunction with the WGS.

The WGS is a collection of higher-level graphic routines, written in PROMAL. It is supplied in both source and object form. The WGS allows you to display graphics in various "windows" on the screen, with automatic "clipping" of lines which would lie outside the window area. It also allows you to draw your graphics using measurement units suited for your application, with automatic scaling. For example, if you wanted to draw a map of North Carolina, you might plot Raleigh at X=372.4 miles and Y=75.28 miles.

If you are a rodent lover, we are sorry to tell you that the GRAPHICS TOOLBOX does not directly support mouse input or Macintosh-style "pull-down" windows; rather, it provides graphics window support, similar to that found on many mainframe graphics packages such as the Tektronix TCS system. Of course, you can easily develop your own pull-down window support with the GTB.

The GRAPHICS TOOLBOX includes a manual and disk with numerous demonstration programs. For only **$29.95 plus $2.50** for shipping and handling ($15.00 outside USA and Canada), you can put the punch of high-resolution graphics into your PROMAL programs. It's fun! □

/FIXES from page 2

Naturally you don't make the patch if you have selected the NOREAL option.

The preceding information applies to the latest version of PROMAL 2.0 as of the publication time of this newsletter. You can find out if you have the latest version of PROMAL by typing the command WHORU from the EXECUTIVE. If the displayed date is 01/08/86 or later, you have the latest version. Otherwise, you may wish to contact SMA to find out how to get your upgrade. □

/FAME from page 5

5. Improvements to the macro assembler provided on Public Domain Disk #1 (Apple and Commodore versions).
6. SID programming on the C-64 with PROMAL.
7. Modifications of S_PRT_IMAGE in the PROMAL GRAPHICS TOOLBOX for popular printers.
8. Useful application programs that would have wide appeal

Now the fine print: We reserve the right to select what is published. And, we can't return submitted disks. □

SYSTEMS MANAGEMENT ASSOCIATES
3325 Executive Drive, P.O. Box 20025
Raleigh, North Carolina 27619
Telephone: (919) 878-3600

Bulk Rate
U.S. Postage
P A I D
Richmond, VA
Permit No. 729